UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/038,473	12/31/2001	Sushma Shrikant Trivedi	04860.P2688	7836

7590 11/05/2008

James C. Scheller
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

EXAMINER	
HUISMAN, DAVID J	

ART UNIT	PAPER NUMBER
2183	

MAIL DATE	DELIVERY MODE
11/05/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/038,473	TRIVEDI ET AL.	
	Examiner	Art Unit	
	DAVID J. HUISMAN	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 August 2008*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-34 and 36-69* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-34 and 36-69* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 December 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 3-34, and 36-69 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment and Extension of Time as received on 8/13/2008.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 1, 5-7, 11-14, 17-20, 23-30, 32-34, 38-40, 44-47, 50-53, 56-63, 65-66, and 68 are rejected under 35 U.S.C. 102(e) as being anticipated by Sazegari, U.S. Patent Number 6,446,198 (herein referred to as Sazegari).

5. Referring to claims 1 and 34 Sazegari has taught a method for execution by a microprocessor in response to receiving a single instruction, the method comprising:

a) receiving a string of bits having a plurality of segments (see Fig.3 and note the permute mask (component 26) has a plurality of segments, i.e., values), receiving a plurality of data elements specifying the plurality of segments in the string of bits (each permute instruction inherently comprises at least an opcode and a register specifier that specifies the mask register. The opcode and the register specifier are data elements which specify the segments in the string of bits. That

is, to the system, the data in the mask register is simply an 80-bit data value. However, the opcode and register specifier indicate to the system that the data in the register specified by the specifier is to be treated as 16 five-bit segments), generating a plurality of indices based on the plurality of data elements, wherein a first one of the plurality of indices is generated from retrieving a first bit segment from the string of bits according to a first one of the plurality of data elements (again, according to the permute opcode and mask register specifier, the system will generate indices from each respective segment in the string of mask bits. Also, see the abstract, figures 3-4, column 2, lines 17-43, and column 4, lines 5-46); Note specifically from column 4, lines 24-38, that 16 simultaneous lookups are performed at multiple locations in multiple lookup tables by extracting 16 five-bit segments among the string of bytes/bits (Fig.3, component 26), where each of the segments is an index into one of the tables). **A second and alternative interpretation of the above limitations is as follows:** receiving a string of bits having a plurality of segments (see Fig.3 and note the permute mask (component 26) has a plurality of segments, i.e., values), receiving a plurality of data elements specifying the plurality of segments in the string of bits (the system receives, via parsing the mask register, a plurality of data elements, i.e., 5-bit elements, each element specifying a segment (mask value) in the string of bits), generating a plurality of indices based on the plurality of data elements, wherein a first one of the plurality of indices is generated from retrieving a first bit segment from the string of bits according to a first one of the plurality of data elements (according to each 5-bit data element, the value specifieid by each element is used to generate an index. Also, see the abstract, figures 3-4, column 2, lines 17-43, and column 4, lines 5-46); Note specifically from column 4, lines 24-38, that 16 simultaneous lookups are performed at multiple locations in multiple lookup tables by

extracting 16 five-bit segments among the string of bytes/bits (Fig.3, component 26), where each of the segments is an index into one of the tables).

b) receiving a configuration indicator, wherein the configuration indicator indicates how to configure a plurality of look-up units into one or more look-up tables for execution of the single instruction, and configuring the plurality of look-up units into one or more look-up tables according to the configuration indicator. See Fig.3, Fig.4, and column 1, lines 36-39. Note that each permute instruction includes two source operand names (data1 and data2). Each operand name represents a plurality of look-up units (16 bytes). Hence, with a pool of N operands to choose from (data1 to dataN), two sets of look-up units will be selected and configured into a look-up table. The configuration indicator is the data specifying the operands (the operand addresses). That is, the permute instruction will be of the form "PERM Dx, S1, S2". S1 and S2 are the look-up units to be configured into one or more look-up tables. S1 and S2 are inherently represented by bits. The groups of bits making up S1 and S2 form the configuration indicator.

c) looking up simultaneously a plurality of entries from the one or more look-up tables using the plurality of indices, the one or more look-up tables configured from the plurality of look-up units, wherein each of said plurality of look-up units is a memory unit that is separate and distinct from others of said plurality of look-up units and is individually accessible independent of operations of the other look-up units. See Fig.4 and column 4, lines 24-32. Note that the table of Fig.4 is divided into two look-up unit memories (data1 and data2). Column 2, lines 17-26, further supports this by saying that a table is logically divided into a number of smaller tables (i.e., multiple logical tables, which are separate and distinct, exist). Furthermore, each unit (data1, data2) is individually accessible independent of operations to other tables. That is,

looking at Figs.3-4, the lookups are independent of one another. All of the indices set forth by the permute mask correspond to either an index from unit data1 or unit data2, not both. An index used to perform a look-up in unit data1 has no effect whatsoever on looking up in unit data2.

d) combining the plurality of entries into a first result (Sazegari abstract, figures 3-4, column 4 lines 5-67);

e) wherein the above operations are performed in response to the microprocessor receiving the single instruction (Sazegari column 4 lines 5-67, and figures 3-4), and note that the operations are performed in response to a single permute instruction.

6. Referring to claims 5 and 38 Sazegari has taught the methods of claims 1 and 34, respectively, further comprising:

receiving a bit pointer, wherein the plurality of segments in the string of bits are determined using the bit pointer and the plurality of data elements. See Fig.3, and column 4, lines 10-11. The bit pointer would be the address of the register used for providing the string of bits. The data elements (byte fields) of that register are then used to determine the segments by holding values corresponding to the segments. So, the register first must be located using a pointer, i.e., address (such as R2 or R5), and then each byte field (data element) of the register is read to determine the segments stored within.

7. Referring to claims 6 and 39 Sazegari has taught the methods of claims 5 and 38, respectively, further comprising:

generating a new bit pointer using the first result. See Fig.3, Fig.4, and column 4, lines 10-11. Note that when the first result is produced, it will be stored in a register (Fig.4, component 38). This register may later be used as a permute mask (Fig.3, component 26).

When the new result is to be used as a mask for the permute instruction, a bit pointer must be generated to address the register holding the first result.

8. Referring to claims 7 and 40 Sazegari has taught the methods of claims 1 and 34, respectively, further comprising:

receiving an offset, wherein the plurality of indices are determined using the offset and the plurality of segments of bits. In order to determine the segments, the register holding the segments must be located. The register is located by supplying a register address (offset) in the permute instruction. A register address is an offset because it dictates the number of registers away the selected register is from register 0. For instance, if register 1 is to be selected as the permute mask, then register address 00001 (assuming 32 registers) would be provided. Address 00001, which corresponds to 1, indicates that the register to be selected (R1) is 1 register away from register 0 (R0). Likewise, if register 5 is to be selected as the permute mask, then register address 00101 (assuming 32 registers) would be provided. Address 00101, which corresponds to 5, indicates that the register to be selected (R5) is 5 registers away from register 0 (R0). The segments of the register 26 (Fig.3) are then used as indices into the lookup tables.

9. Referring to claims 11 and 44 Sazegari has taught the methods of claims 1 and 34, respectively, wherein the single instruction specifies a total number of entries contained in each of the one or more look-up tables. See column 4, lines 33-38. Note that if the table size is 32, then 5 bits are supplied in the instruction to address any one of the 32 entries. Since the instruction specifies a 5-bit value, and the 5-bit value specifies that there are 32 entries in the table, then it follows that the instruction specifies the total number of entries in the table. If

applicant is suggesting that the size of the table is unknown prior to the instruction specifying an explicit size, then this concept should be claimed.

10. Referring to claims 12 and 45 Sazegari has taught the methods of claims 11 and 44, respectively, wherein the total number of entries is one of:

a) 256 (Sazegari column 3 lines 57-58, column 2 lines 35-43, column 2 lines 17-25);

b) 512; and

c) 1024.

11. Referring to claims 13 and 46 Sazegari has taught the methods of claims 1 and 34, respectively, wherein the single instruction specifies a total number of bits used by each entry contained in the one or more look-up tables. See Fig.3 and note that each permute instruction specifies that each entry uses 8 bits because the permute instruction indicates that 16 values are to be loaded into the result register (Fig.3). Since the result register is 128 bits wide, it follows that each value is 8-bits wide. So, the instruction indicates the bits used by each entry since the number of loads and the size of the register to be loaded are known.

12. Referring to claims 14 and 47 Sazegari has taught the methods of claims 13 and 46, respectively, wherein the total number of bits is one of:

a) 8 (Sazegari column 3 lines 57-58, column 2 lines 35-43, column 2 lines 17-25);

b) 16; and

c) 24.

13. Referring to claims 17 and 50 Sazegari has taught the methods of claims 1 and 34, respectively, wherein said combining the plurality of entries comprises:

selecting a valid data from the plurality of entries. See Fig.3 and Fig.4, and note that data is selected and combined into a single result. Entries that are read are assumed to be valid. Note that the selecting and combining occur in response to the permute instruction alone (in the case of Fig.4).

14. Referring to claims 18 and 51 Sazegari has taught the methods of claims 17 and 50, respectively, further comprising:

generating an indicator indicating whether none of the plurality of entries is valid. Entries that are not read are assumed to be invalid, i.e., they do not hold data that is required by the instruction. These entries are indicated by indicating the valid entries (if x+y=z, and you know x and z, then you also know y).

15. Referring to claims 19 and 52 Sazegari has taught the methods of claims 17 and 50, respectively, wherein the valid data is selected according to priorities of the one or more look-up tables from which the plurality of entries are looked up. See Fig.3 and Fig.4 and note that more lookups need to be performed in data2. Therefore, it requires more attention and is given more attention than data1.

16. Referring to claims 20 and 53 Sazegari has taught the methods of claims 17 and 50, respectively, wherein said combining the plurality of entries further comprises:

formatting the valid data according to a type of the valid data. The valid data, when read, are stored into byte fields of the result register. The format is to have 16 bytes of data read and stored.

17. Referring to claims 23 and 56 Sazegari has taught the methods of claims 1 and 34, respectively, wherein an entry in the plurality of entries contains:

a) information indicating whether the entry is valid (Sazegari column 4 lines 63-67; the bits indicate which of the entries will be in the final result, which makes them valid).

b) information indicating a type of the entry; and

c) information indicating a number of bits of a code word to be decoded.

18. Referring to claims 24 and 57 Sazegari has taught the methods of claims 1 and 34, respectively, wherein the string is received from an entry in a register file (Sazegari Figure 3).

19. Referring to claims 25 and 58 Sazegari has taught the methods of claims 24 and 57, respectively, wherein the single instruction specifies an index of the entry in the register file (Sazegari abstract, figures 2-3, and 5, column 4 lines 5-67).

20. Referring to claims 26 and 59 Sazegari has taught the methods of claims 1 and 34, respectively, further comprising:

receiving a first number indicating a position of a last bit of input in the string of bit. The register address of the permute mask (Fig.3) would be such number since it indicates that the input string ends with the last bit in the selected register.

21. Referring to claims 27 and 60 Sazegari has taught the methods of claims 26 and 59, respectively, further comprising:

generating an indicator indicating whether any bit after the last bit of input is used in obtaining the first result. The signal to cause a write to occur to register 38 (Fig.4) is an indicator that bits encountered subsequent to the string of bits need to be written to the register in order to obtain the result.

22. Referring to claims 28 and 61 Sazegari has taught the methods of claims 12 and 45, respectively, further comprising:

generating an indicator indicating whether one of the plurality of segments of bits contains a predetermined code. The opcode for the instruction is an indicator that each segment will contain an 8-bit code used to index into lookup tables.

23. Referring to claims 29 and 62 Sazegari has taught the methods of claims 28 and 61, respectively, wherein the predetermined code represents an end of block condition. The opcode of the permute instruction indicates that the end of the input block occurs with the last amount of data in the register selected by the permute instruction as the permute mask.

24. Referring to claims 30 and 63 Sazegari has taught the methods of claims 1 and 34, respectively, further comprising:

receiving at least one format;

formatting the string of bits into at least one escape data according to the at least one format; and

combining the at least one escape data and the first result into a second result. See Fig.3 and note that in addition to a permute mask 26, register operands (28, 30) are also be used. So, suppose that a permute instruction is executed, a table-memory lookup occurs and a first result is stored in register 28 (this would correspond to the operation shown in Fig.4 with the result being stored in register 38). Then when a second permute instruction is executed, the string of bits in the permute mask are formatted into 16 groups of 8-bits (this is the format of the mask). Then, if one of the input registers 28 is the register in which the first result was stored, then the combination of mask and first result (and second operand register) would yield a second result 32.

25. Referring to claims 32 and 65 Sazegari has taught the methods of claims 30 and 63, respectively, wherein the at least one format is received from an entry of a register file (Sazegari abstract, figures 2-3, and 5, column 2 lines 17-43, column 3 lines 40-46, column 4 lines 5-46; column 2 lines 17-25).

26. Referring to claims 33 and 66 Sazegari has taught the methods of claims 32 and 65, respectively, further comprising: wherein the single instruction specifies an index of the entry in the register file (Sazegari abstract, figures 2-3, and 5, column 2 lines 17-43, column 3 lines 40-46, column 4 lines 5-46; column 2 lines 17-25).

27. Referring to claim 68, Sazegari has taught a method as described in claim 1. Sazegari has further taught that each of the plurality of indices corresponds to a different one of the plurality of lookup tables. Dictionary.com defines a table as "an orderly arrangement of data, especially one in which the data are arranged in columns and rows in an essentially rectangular form." Consequently, looking at Fig.4 of Sazegari, memory 34, may be viewed as 1 table of 32 entries, 2 tables of 16 entries each, 4 tables of 8 entries each, and so on, until you have 32 tables of 1 entry each (1 row, 8 columns). Since there are 16 indices per permute instruction and 32 entries in the memory, they may each map to a different table if each of the indices is unique. For instance, if the memory is viewed as 32 tables (1 row/8 columns each), then they will all map to a different table provided the indices are unique. Or, if all the indices were odd numbers or all even numbers, then they would each map to a different table among 16 tables (2 rows/8 columns each).

Claim Rejections - 35 USC § 103

28. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

29. Claims 3-4, 8-10, 15-16, 21-22, 31, 36-37, 41-43, 48-49, 54-55, 64, and 69 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sazegari in view of the examiner's taking of Official Notice.

30. Referring to claims 3 and 36 Sazegari has taught the methods of claims 1 and 34, respectively, wherein the plurality of data elements are received from an entry in a register file. See Fig.3, component 26, and column 4, lines 10-11. Sazegari has further taught a media processor and a memory controller (if the system has memory (RAM), then it also has a memory controller, i.e., anything that controls/manages RAM in some way) but has not explicitly taught that they are integrated on a single integrated circuit. However, as shown in In re Larson 144 USPQ 347 (CCPA 1965), to make integral is generally not given patentable weight or would have been an obvious improvement. Integrated circuits allow for high-speed communication between components since everything is on a single chip (additional/longer wires/buses connecting multiple chips are unnecessary). ICs also have low power dissipation and reduced manufacturing costs compared with board-level integration. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Sazegari such that the processor and memory controller are integrated.

31. Referring to claims 4 and 37 Sazegari, as modified, has taught the methods of claims 3 and 36, respectively, wherein the single instruction specifies an index of the entry in the register file (Sazegari abstract, figures 2-3, and 5, column 4 lines 5-67).

32. Referring to claims 8 and 41 Sazegari has taught the methods of claims 1 and 34, respectively, further comprising:

a) partitioning look-up memory into the one or more look-up tables before said looking-up (Sazegari column 2 lines 17-25);

b) Sazegari has not explicitly taught that the microprocessor is a media processor formed on a monolithic integrated circuit. However, Official Notice is taken that processors being formed on an integrated circuit is well known and expected in the art. ICs allow for low power dissipation and lower manufacturing costs. Furthermore, as shown in In re Larson 144 USPQ 347 (CCPA 1965), to make integral is generally not given patentable weight or would have been an obvious improvement. As a result it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Sazegari such that the processor is formed on a monolithic integrated circuit.

33. Referring to claims 9 and 42 Sazegari, as modified, has taught the methods of claims 8 and 41, respectively, wherein the look-up memory comprises a plurality of look-up units, and wherein said partitioning look-up memory comprises configuring the plurality of look-up units into the one or more of look-up tables (Sazegari column 3 lines 57-58, column 2 lines 35-43, column 2 lines 17-25).

34. Referring to claims 10 and 43 Sazegari, as modified, has taught the methods of claims 9 and 42, respectively, further comprising: wherein each of the plurality of look-up units comprises 256 8-bit entries (Sazegari column 3 lines 57-58, column 2 lines 35-43).

35. Referring to claims 15 and 48 Sazegari, as modified, has taught the methods of claims 8 and 41, respectively, wherein the one or more look-up tables are configured according to an indicator in an entry in a register file (Sazegari column 3 lines 57-58, column 2 lines 35-43, column 2 lines 17-25).

36. Referring to claims 16 and 49 Sazegari, as modified, has taught the methods of claims 15 and 48, respectively, wherein the single instruction specifies an index of the entry in the register file (Sazegari abstract, figures 2-3, and 5, column 4 lines 5-67).

37. Referring to claims 21 and 54 Sazegari has taught the methods of claims 20 and 53, respectively. Sazegari has not taught that the type of the valid data is one of:

a) zero fill;

b) sign magnitude; and

c) two complement.

However, Official Notice is taken that it is well known in the art to represent binary data in two's complement form. Such a form allows the system to efficiently represent negative numbers as well as positive numbers. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Sazegari to include two's complement data.

38. Referring to claims 22 and 55 Sazegari, as modified, has taught the methods of claims 21 and 54, respectively. Sazegari has not taught retrieving a sign bit from the string of bits for the valid data, wherein the first result is obtained by formatting the valid data using the sign bit when

the type of the valid data is sign magnitude. However, Official Notice is taken that sign magnitude notation is well known in the art. This format allows for straightforward way to represent positive and negative numbers. The format choice is really nothing more than a designer choice. Different designers choose different formats for different reasons. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Sazegari to include a sign magnitude format, which would include the specifics set forth above.

39. Referring to claims 31 and 64 Sazegari has taught the methods of claims 30 and 63, respectively. Sazegari has not taught that the at least one format is for data of a type which is one of:

a) zero fill;

b) sign magnitude; and

c) two complement.

However, Official Notice is taken that it is well known in the art to represent binary data in two's complement form. Such a form allows the system to efficiently represent negative numbers as well as positive numbers. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Sazegari to include two's complement data.

40. Referring to claim 69, Sazegari has taught a method as described in claim 1. Sazegari has not taught that each of the plurality of look-up tables is larger than a vector register. However, as shown in In re Rose, 105 USPQ 237 (CCPA 1955) changes in size/range are generally not given patentable weight or would have been obvious improvements. Clearly, the size of the lookup tables may be increased to hold more data. This would cause the amount of possible indices to

grow and therefore, the instruction would also have to be modified to include fewer indices. But, the general idea still remains. Simultaneous lookups to multiple tables is still taught by Sazegari. Smaller tables may be used, bigger tables may be used, and other features may be adjusted to accommodate the changed sizes of the tables, but the process of simultaneously looking up still remains the same. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Sazegari such that each of the lookup tables is larger than a vector register.

41. Claim 67 is rejected under 35 U.S.C. 103(a) as being unpatentable over Sazegari in view of Shams, U.S. Patent No. 5,526,501.

42. Referring to claim 67, Sazegari has taught a method as described in claim 1. Sazegari has not taught that the plurality of segments of bits in the string of bits are of variable lengths. However, Shams has taught multiple lookup tables of varying size. See column 3, line 49, to column 4, line 9. Consequently, the segments that are used to index into the tables would also be of varying size. By having tables of varying size, more or less data may be stored in any given table. Processes which involve a lot of data of one type may store that data in a larger table whereas if it involves much less data of another type, the data may be stored in a smaller table, thereby leaving a bigger table for an application that needs it. In addition, the size of the tables is not critical when it comes to Sazegari. Sazegari merely teaches making simultaneous lookups from multiple tables. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Sazegari in view of Shams such that multiple tables of varying size are included, and consequently, multiple segments of varying size are included.

Response to Amendment

43. Applicant is respectfully requested to direct the examiner to a specific section of the specification, and to a particular drawing, which supports applicant's most recent amendment to claims 1 and 34 filed on August 13, 2008. The examiner is not afforded the time to search through 160 pages of disclosure and through 101 figures to try and find support for the amendment. Hence, to help the examiner better locate and understand the inventive concept, applicant is asked to show support for the amendment. Failure to do so may result in a new matter rejection under 35 U.S.C. 112, 1st paragraph, in the next office action.

Response to Arguments

44. Applicant's arguments filed on August 13, 2008, have been fully considered but they are not persuasive.

45. Applicant argues the novelty/rejection of claims 1 and 34 on page 18 of the remarks, in substance that:

> "Sazegari merely discloses identifying each of the entries in the look-up table with five bits of each byte in the index register. In contrast, amended claim 1 refers to receiving a plurality of data elements specifying the plurality of segments in the string of bits; and generating a plurality of indices based on the plurality of data elements, wherein a first one of the plurality of indices is generated from retrieving a first bit segment from the string of bits according to a first one of the plurality of data elements."

46. These arguments are not found persuasive for the following reasons:

a) As discussed in the rejections above, Sazegari anticipates applicant's claims 1 and 34 under two different interpretations:

- The plurality of data elements may be considered to be the permute instruction components, i.e., the opcode and the register specifiers. Hence, it is in response to the opcode and to the specification of the mask register that the system knows where to get the mask data and how to divide the mask data up (i.e., into 5-bit segments). This division is built into the functionality of the permute opcode.
- The plurality of data elements may be considered to be the 5-bit data elements individually retrieved from the mask register. These five-bit elements specify the segments (mask values) and are used to generate indices.

Conclusion

47. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DAVID J. HUISMAN whose telephone number is (571)272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/David J. Huisman/
Primary Examiner, Art Unit 2183
October 1, 2008